

SISTEMA INTELIGENTE PARA GERENCIAMENTO DA COLHEITA DO MEXILHÃO *Perna perna* (Linné, 1758)

Cássio Veiga Ramos

cassioo@terra.com.br

Anita Maria da Rocha Fernandes, Dra

anita@inf.univali.br

Curso de Ciências da Computação – Centro de Ciências Tecnológicas, da Terra e do Mar- CTTMar
Universidade do Vale do Itajaí – UNIVALI – Santa Catarina – Brasil

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de informação *on-line* que auxilie na tomada de decisão quanto às condições de colheita do Mexilhão *Perna perna*. O sistema parte de um projeto já desenvolvido, intitulado Sistema Especialista para Determinação do Índice de Condição de Colheita do Mexilhão *Perna perna* (Linné, 1758) – SEDICM. Este projeto teve como objetivo geral avaliar as condições ambientais e biológicas, para auxílio na tomada de decisão quanto à colheita do Mexilhão para a estação experimental de maricultura da UNIVALI, Campus V, Penha, Santa Catarina. O SEDICM contém um banco de dados com informações sobre o ambiente onde a área de cultivo está inserida, sendo que um sistema especialista foi implementado para o auxílio na tomada de decisão. O banco de dados foi desenvolvido no Delphi 3.0 e o sistema especialista no Expert SINTA. Porém, o SEDICM apresentou problemas relativos a ergonomia de sua interface e limitações de execução, devido à plataforma onde foi desenvolvido. O principal objetivo do projeto atual foi disponibilizar na internet toda funcionalidade do SEDICM, implementando melhoras ergonômicas e adicionando informações que foram absorvidas no decorrer do desenvolvimento deste projeto. Para tanto, o sistema foi implementado em Java e utilizou a *shell* Jess para implementar o sistema especialista.

Palavras Chave: Inteligência Artificial; Sistemas Especialistas; Sistemas de Informação, JESS.

ABSTRACT

This document presents the development of an on-line information system that helps on the decision making about the harvesting of the *Perna perna* mussel crop. The system source is a already existing project called Expert System for Determination of the condition rate *Perna perna* (Linné, 1758) mussel crop – SEDICM. The main goal of this project was to evaluate the biological and environmental condition, to help on the decision making on the harvesting of the mussel crop for the Univali Campus V – Sea Culture Experimental Station, in Penha – SC. The SEDICM has a data base with information about the environment where the harvesting is taking place and an expert system was implemented to assist on the decision making. The data base was developed on Delphi 3.0 and the expert system on the Expert Sinta. But, SEDICM presented problems related to interface and execution limitations from the environment it was developed. The main goal of the actual project was to have available on the internet all the functioning of the SEDICM, implementing ergonomic improvements and adding information that was acquired during the development of this project. For this, the project was implemented on Java and used a Jess for implementing the expert system.

Key Words: Artificial Intelligence; Expert System; Information System, JESS.

CACIC'2003 - Congresso

1. Introdução

Este projeto disponibiliza na internet as informações necessárias para que os maricultores em geral, possam ter uma ferramenta que os auxilie na escolha do melhor período para colheita do mexilhão.

Um sistema de informação *on-line* foi criado, tendo como base o SEDICM – Sistema Especialista para Determinação do Índice de Colheita do Mexilhão *Perna perna*, que forneceu toda base de regras necessárias para escolha do melhor período para colheita do mexilhão. Um dos principais objetivos foi melhorar o protótipo proposto no projeto anterior, principalmente sob aspectos ergonômicos e funcionais.

O sistema de informação implementado, foi desenvolvido em Java, linguagem de programação amplamente utilizada na internet. Alguns aspectos foram levados em consideração para a escolha da linguagem de programação: Java é uma linguagem completamente orientada a objetos; Java é baseada nas duas linguagens de programação mais utilizadas no mundo atualmente, C e C++; Algumas funcionalidades mais complexas e propensas a erros de C++ foram extraídas, como ponteiros, gabaritos, sobrecarga de operador e múltipla herança. Fato que possibilita incrível portabilidade de Java em relação aos sistemas operacionais existentes atualmente. Essa portabilidade resulta em economia de tempo e redução de custos no desenvolvimento de sistemas (DEITEL & DEITEL, 2003).

Para implementar o sistema especialista, foi utilizada a ferramenta *shell* Jess. Como o Jess é uma classe de Java, a integração foi total entre a linguagem de programação e a ferramenta *shell*.

2. Fundamentação Teórica

2.1 Sistemas de Informação

Sistema de Informação (SI) é um conjunto organizado de pessoas, hardware, software, redes de comunicações e recursos de dados que coleta, transforma e dissemina informações em uma organização.

Segundo O'BRIEN (2001), há uma grande mudança na tecnologia da informação cuja importância os executivos acadêmicos e tecnólogos concordam com unanimidade. É o crescimento explosivo da Internet e das tecnologias e aplicativos afins e seu impacto nos negócios, sociedade e sobre a própria tecnologia da informação. Assim como O'BRIEN (2001), LAUDON & LAUDON (1999) citam que a Internet está mudando o modo como as empresas são operadas e as pessoas trabalham e como a tecnologia da informação apóia as operações empresariais e as atividades de trabalho do usuário final.

A Internet se tornou uma plataforma vital de telecomunicações para comunicações eletrônicas e para a colaboração e o comércio eletrônico entre as empresas e seus funcionários, clientes, fornecedores e parceiros comerciais. *Sites* comerciais na Internet se tornaram casas de compensação para a troca interativa de informações por e-mail, sistemas de *chat*, fóruns de discussão e edição de multimídias. Os *sites* das empresas também servem como pontos de atacado e varejo eletrônico para compra e venda de uma ampla variedade de produtos e serviços (O'BRIEN, 2001; LAUDON & LAUDON, 1999).

2.2 Sistemas Especialistas

Segundo LIEBOWITZ (1994) e STAIR & REYNOLDS (1999), os Sistemas Especialistas (SE's) são programas de computadores que imitam o comportamento de especialistas humanos dentro de um domínio de conhecimento específico. Estes programas são particularmente relevantes para realização de inferências e deduções em cima de problemas que envolvem aspectos não estruturados. Assim como os especialistas humanos, estes sistemas usam lógica simbólica e heurística para encontrar soluções para problemas e, da mesma forma, podem cometer erros.

Os sistemas especialistas têm sido utilizados também para determinar o valor do limite dos cartões de crédito. As pesquisas realizadas em IA nas últimas duas décadas estão resultando em SE's que exploram novas oportunidades de negócios, incremento total de lucratividade, redução de custos e que proporcionam melhores serviços para os clientes (STAIR & REYNOLDS, 1999).

Um dos maiores problemas com os SE's estava na extração de informações dos especialistas. Em primeiro lugar, se os especialistas estavam dispostos a repassar as informações, rapidamente podia-se descobrir que seus conhecimentos não seriam facilmente interpretados. Pode-se dizer ainda que sem uma adequada base de conhecimento, o sistema não pode dispor resultados satisfatórios (WHALEY, 2002).

2.3 JESS – The JAVA Expert System Shell

Jess é uma *shell* para construção de sistemas especialistas, que foi desenvolvida na linguagem Java. Esta *shell* suporta o desenvolvimento de sistemas especialistas baseados em regras que podem ser facilmente integrados em programas escritos em Java (JESS, 2003).

A versão atual do Jess é a 6.0, compatível com todas versões de Java, a partir de Java 1.2. Jess é uma biblioteca escrita em Java e é muito similar com outra *shell* para sistemas especialistas: CLIPS. Esta linguagem possui uma *applet* genérica, a classe `jess.ConsoleApplet`, que pode ser usada, em geral, em situações simples baseadas em perguntas e respostas (*ibidem*).

Conforme JESS (2003), esta linguagem de programação possui propósito geral e é muito eficaz por ter acesso direto a todas classes e bibliotecas da linguagem Java. Porém, enquanto código escrito em Java precisa ser compilado antes para que um interpretador possa executar o programa, um comando de Jess é executado imediatamente após sua digitação. Isto permite testar as APIs de Java interativamente e auxiliar expressivamente no desenvolvimento de novos programas.

Esta *shell* atualmente é tão rápida quanto algumas *shells* populares desenvolvidas através da linguagem de programação C, especialmente em problemas de larga escala, onde a performance desses sistemas é determinada pela qualidade dos algoritmos. A máquina de inferência de Jess, utiliza o conhecido algoritmo de Rete. Este algoritmo trabalha de maneira a negociar espaço por tempo. Jess contém alguns comandos que permitem sacrificar um pouco a performance do sistema para diminuir o uso de memória. Apesar de tudo, a memória usada por esta linguagem é pequena, e programas de tamanho moderado facilmente se enquadram no padrão de uso de memória de Java, que é de 16 Mb (JESS, 2003).

2.3.1 O Algoritmo de RETE

Conforme GIARRATANO (1989) *apud* FERNANDES (1995) e JESS (2003), as linguagens baseadas em regras, como por exemplo JESS, CLIPS, ART, OPS5 e OPS83, usam um algoritmo

bastante eficiente para unificação dos padrões nas regras, a fim de determinar quais regras têm seus antecedentes satisfeitos. Este algoritmo é chamado de algoritmo de Rete.

O propósito de Jess é aplicar continuamente um conjunto de regras num determinado conjunto de dados (base de conhecimento). Pode-se definir as regras necessárias para o sistema especialista que estiver sendo desenvolvido.

Tem-se solução simples para o problema se o processo de unificação de regras tiver que acontecer apenas uma vez. A máquina de inferência examina cada regra e, então, procura o conjunto de fatos que determinam se os padrões da regra serão satisfeitos. Porém, em linguagens baseadas em regras, o processo de unificação ocorre repetidamente. Assim, a base de conhecimento sofre alterações continuamente. Normalmente, a lista de fatos será modificada durante cada ciclo de execução. Novos fatos podem ser acrescentados à lista de fatos ou fatos antigos podem ser removidos desta lista e o conjunto de regras que são satisfeitas devem ser mantidas e atualizadas. Com estas mudanças, padrões que não foram satisfeitos anteriormente podem ser satisfeitos ou padrões satisfeitos anteriormente podem não ser mais satisfeitos (FERNANDES, 1995; JESS, 2003).

Após a máquina de inferência ter checado cada regra para direcionar a busca dos fatos, uma técnica é fornecida para solucionar este problema. Embora novos fatos são adicionados e fatos antigos são excluídos a todo instante, a percentagem de fatos que são alterados por unidade de tempo é relativamente pequena. Como a percentagem de fatos alterados é pequena a cada ciclo de execução, apenas uma pequena porcentagem de regras é afetada. Um tempo desnecessário é gasto para a busca dos fatos necessários, pois as regras irão percorrer toda a lista de fatos a cada ciclo de execução. A lista de fatos possui duas ramificações: *left-hand-side* (LHS) e *right-hand-side* (RHS). O processo das regras encontrarem os fatos, *rules findig facts*, tem uma complexidade computacional na ordem de $O(RF^P)$, onde R é o número de regras, P é o número médio de padrões por regra LHS, e F é o número de fatos na base de conhecimentos. Esta complexidade aumenta drasticamente com o incremento de padrões por regras (*ibidem*).

Segundo FORGY (1985) *apud* FERNANDES (1995) o algoritmo de unificação de padrões de Rete foi projetado para resolver este problema de redundância temporal que existe em um SE baseado em regras.

O algoritmo de Rete salva o estado do processo de unificação ciclo por ciclo e recalcula as mudanças de um determinado estado apenas para aquelas que ocorreram na lista de fatos. Ou seja, se um conjunto de padrões encontra dois dos três fatos solicitados em um ciclo, uma checagem não precisa ser feita no próximo ciclo para dois fatos que já foram encontrados. O terceiro fato que é interessante. O estado do processo de unificação é atualizado apenas se fatos são acrescentados ou removidos. O processo de unificação será rápido quando o número de fatos acrescentados ou removidos for pequeno se comparado ao total de fatos e padrões. Já, se nenhum dos fatos for alterado, então o processo de unificação trabalha unificando todos os fatos com todos padrões (FERNANDES, 1995).

O processo de unificação pode apresentar dois tipos de informação de estado distintos: a unificação parcial e a unificação padrão. Uma unificação parcial para uma regra é qualquer conjunto de fatos que satisfazem os padrões da regra. Sendo assim, uma regra com três padrões terá unificações parciais para o primeiro padrão, o primeiro e segundo padrões, o primeiro, segundo e terceiro padrões, e assim por diante. Uma unificação padrão ocorre quando um fato satisfaz um padrão simples em qualquer regra, não considerando as variáveis em outros padrões que possam restringir o processo de unificação (*ibidem*).

2.4 O Sistema SEDCIM

O SEDICM - Sistema Especialista para a Determinação do Índice de Condição de Colheita do Mexilhão *Perna perna* (Linné, 1758), foi desenvolvido para a estação experimental de maricultura da Universidade do Vale do Itajaí – UNIVALI, com a finalidade de obter o aumento da produção de marisco e um maior controle da qualidade do produto final (SÁ, 1999).

O sistema é composto por um banco de dados com informações sobre o ambiente onde a área de cultivo está inserida e pelo sistema especialista propriamente dito. O banco de dados foi desenvolvido no Delphi 3.0 e o sistema especialista na *shell* Expert SINTA.

2.4.1 Etapas no desenvolvimento do SEDICM

Segundo SÁ (1999), algumas das etapas que foram seguidas para o desenvolvimento do SEDICM foram: Estabelecimento do Problema a ser Resolvido; Aquisição do Conhecimento; Projeto do Sistema; Desenvolvimento do Sistema.

O processo de coleta e armazenamento dos dados, bem como a tomada de decisão quanto às condições de colheita são realizados manualmente, devido a escassez de software desta natureza no mercado. O pesquisador coleta a amostra, colhe os dados, armazenando-o em planilhas eletrônicas, as quais geram gráficos para a análise da produção e variação do Índice de Condição (IC).

Entretanto, esta análise das condições torna-se muito trabalhosa, pois depende da interpretação cuidadosa de muitas variáveis, conseqüentemente podendo ocasionar erros e tornando-se uma análise muito lenta. Por todos estes fatores torna-se muito difícil obter-se um diagnóstico exato sobre as condições de colheita. Através do cruzamento de uma série de variáveis, apresentadas sob a forma de gráficos ou planilhas, é tomada a decisão para colheita do mexilhão *Perna perna*.

A aquisição do conhecimento para o SEDICM foi realizada em três etapas: levantamento bibliográfico sobre o problema em questão; entrevistas periódicas com o especialista; acompanhamento de campo. O levantamento bibliográfico foi feito com base em manuais de cultivo fornecidos pela EPAGRI, apostilas do curso de Oceanografia da UNIVALI, e relatórios de pesquisa dos projetos desenvolvidos junto à comunidade de maricultores do município de Penha.

As entrevistas periódicas foram feitas ao longo de três meses. Estas entrevistas serviram para melhor compreensão do problema em questão e especificação das principais variáveis a serem utilizadas pelo sistema.

O acompanhamento de campo foi feito durante o mês de Setembro de 1998, no Campus V da UNIVALI, no município de Penha. Este acompanhamento serviu para analisar, na prática, como todo o processo de cultivo ocorre. Como produto deste acompanhamento teve-se a determinação de mais algumas variáveis a serem incluídas no sistema. Cabe ressaltar que, o processo de aquisição de conhecimento foi realizado com muita eficiência por SÁ (1999). Com habilidade, conseguiu definir as regras e padrões que o sistema especialista deverá seguir para as tomadas de decisão.

O projeto do sistema SEDICM foi elaborado em duas etapas: representação do conhecimento e modelagem do sistema.

Após a aquisição do conhecimento, SÁ (1999) iniciou a representação deste conhecimento através da determinação das variáveis e seus respectivos valores; determinação das variáveis objetivo do sistema; determinação das regras de produção.

O SEDICM foi desenvolvido na *shell* Expert SINTA, o qual efetua o teste ambiental, associando as variáveis que influenciam na tomada de decisão. Ele é composto por dez variáveis:

- ♣ Temperatura: classificada em três valores (alta, média, baixa);
- ♣ Salinidade: classificada em três valores (alta, média, baixa);
- ♣ Transparência: classificada em três valores (alta, média, baixa);
- ♣ Estação do ano: classificada em quatro valores (primavera, verão, outono, inverno);
- ♣ Intensidade do vento: classificada em três valores (fraco, fraco-moderado, moderado-forte);
- ♣ Direção do vento: classificada em três valores (sul, nordeste, oeste);
- ♣ Ondas: classificada em cinco valores (sem ondas, ondas pequenas, ondas médias, ondas grandes, ondas muito grandes);
- ♣ Correnteza: classificada em cinco valores (0,2 m/s; 0,4 m/s; 0,6 m/s; 0,8 m/s; mais de 0,8 m/s);
- ♣ Coliformes: classificada em cinco valores (0 células por 100ml, menor que 70 cel/ml, 70 cel/ml, 70-100 cel/ml, mais de 100 cel/ml);
- ♣ Maré: classificada em cinco valores (enchendo, cheia, parada, vazante, baixa).

O SEDICM possui uma variável objetivo chamada de Diagnóstico, classificada em quatro valores: Ótimo, Bom, Regular e Ruim.

As variáveis do sistema são combinadas em pares, disparando cinco variáveis intermediárias (condição 1, condição 2, condição 3, condição 4, condição 5), que possuem quatro valores (ótimo, bom, regular, ruim), a partir da associação das cinco condições obtém-se o diagnóstico, classificado em quatro valores (ótimo, bom, regular, ruim).

- ♣ Condição 1: Temperatura e salinidade;
- ♣ Condição 2: Direção do vento e intensidade do vento;
- ♣ Condição 3: Transparência e estação do ano;
- ♣ Condição 4: Ondas e correnteza;
- ♣ Condição 5: Maré e coliformes;

Para estabelecer as regras a serem utilizadas no sistema foi necessário modelar o conhecimento adquirido em forma de árvores de decisão. A base de conhecimento do sistema possui um conjunto de 140 regras de produção. A seguir tem-se alguns exemplos destas regras.

Exemplo 1:

SE Temperatura = Média E Salinidade = Alta

ENTÃO Condição 1 = Bom CNF 100%

Exemplo 2:

SE Direção do Vento = Nordeste E Intensidade do Vento = Fraco-Moderado

ENTÃO Condição 2 = Ótimo CNF 100%

Exemplo 3:

SE Transparência = Média E Estação do Ano = Inverno

ENTÃO Condição 3 = Bom CNF 100%

Exemplo 4:

SE Ondas = Sem Ondas E Correnteza = 0,2 m/s

ENTÃO Condição 4 = Bom CNF 100%

Exemplo 5:

SE Maré = Cheia E Coliformes = Menor que 70 células/100 ml

ENTÃO Condição 5 = Bom CNF 100%

Exemplo 6:

SE Condição 1 = Bom E Condição 2 = Ótimo E Condição 3 = Bom E Condição 4 = Bom E Condição 5 = Bom

ENTÃO Diagnóstico = Bom CNF 100%

3. O Sistema Proposto

O sistema foi desenvolvido para ser acessado via internet e é independente de plataforma. Qualquer usuário que possua um programa para acessar páginas na internet (*browser*), poderá utilizá-lo. Possui quatro módulos: cadastro, amostra ambiental, amostra biológica e relatórios.

No módulo cadastro, o usuário pode realizar o cadastramento da amostra ambiental, amostra biológica, área de cultivo, pesquisador e produtor, sendo que as funções de consulta, alteração e exclusão também estarão disponíveis. As informações referentes à amostra ambiental, determinam qual o diagnóstico para colheita do mexilhão. Estes dados são coletados do meio ambiente, e atualmente, para encontrar o diagnóstico, são utilizadas planilhas, o que dificulta o trabalho do usuário e proporciona o surgimento de erros, visto que é feito manualmente.

Quanto ao cadastramento da amostra biológica, é de suma importância para o acompanhamento da produção e controle da qualidade dos mexilhões. São utilizadas fórmulas matemáticas para determinação dos índices de condição comercial, biológico e de rendimento. Este trabalho, atualmente, também é realizado manualmente. O cadastramento da área de cultivo, pesquisador e produtor, são de caráter informativo e servem para um melhor controle do sistema.

No módulo amostra ambiental, está implementada a principal funcionalidade do sistema. É neste módulo que é realizado o cálculo diagnóstico. Para isso, o usuário escolhe qual amostra ambiental previamente cadastrada, que será submetida ao cálculo. O sistema captura todas informações que foram registradas no banco de dados referente à amostra que foi escolhida e envia as informações para o sistema especialista. Este foi implementado utilizando a *shell* Jess (*Java Expert System Shell*). O resultado é o diagnóstico para colheita do mexilhão, que poderá ser ótimo, bom, regular e ruim. Mais adiante será detalhada a maneira com que cada passo foi implementado afim de encontrar o diagnóstico.

Assim como o módulo amostra ambiental solicita ao usuário a escolha de qual amostra será submetida ao cálculo, também o módulo amostra ambiental pede qual amostra deverá ser lida do banco de dados para efetuar os cálculos que determinarão os índices de condição comercial, biológico e de rendimento.

Por fim foi disponibilizado para o usuário relatórios cadastrais e relatórios de acompanhamento das amostras, que buscam no banco de dados as informações que foram armazenadas.

3.1 Implementação

O sistema de informações via *web* que foi criado para colocar em prática este projeto, possui quatro módulos: cadastros, amostra ambiental, amostra biológica e relatórios. O primeiro passo realizado foi a criação das tabelas no banco de dados MySQL. Na sequência, foi implementado o sistema especialista, utilizando a *shell* Jess. Para isso, foram criadas a base de regras e um pacote escrito em Java, contendo uma classe para integrar o Jess com a linguagem de programação Java.

O sistema *web* é uma combinação de *tags* HTML com *tags* JSP. As *tags* HTML foram utilizadas para criação da parte estática ou visual do sistema. Utilizou-se a ferramenta Dreamweaver 4.0 da Macromedia para isso. O código HTML que foi gerado por essa ferramenta, foi editado com intenção de integrar as *tags* JSP necessárias para criação da parte dinâmica do sistema.

Os objetos modelados no diagrama de objetos foram transformados em tabelas, para que as informações referentes aos mesmos pudessem ser armazenadas. O banco de dados utilizado foi o MySQL, que suporta a linguagem SQL para sua manipulação.

Para criação das tabelas, inicialmente, criou-se um nome para o banco de dados e em seguida este foi selecionado, para que pudesse ser manipulado. Para isso, no diretório de instalação do MySQL, \mysql, o comando mysql acessa o prompt do banco de dados MySQL.

Com isso as tabelas para o sistema puderam ser criadas. A seguir os scripts que foram utilizados para cada tabela, onde o comando SQL “create” cria a tabela com os parâmetros que se seguem. Todas as tabelas foram criadas utilizando a ferramenta SQLYog. Esta ferramenta visual, facilitou a criação das tabelas, visto que a utilização do banco através do prompt do MySQL não traz facilidades.

Este projeto possui implementado um sistema especialista (SE) que possibilita o gerenciamento da colheita dos mexilhões, onde a melhor época para colheita pode ser estabelecida através desta funcionalidade do sistema *web*. O sistema especialista encontra-se no módulo Amostra Ambiental do sistema, e utiliza a *shell* Jess. Basicamente, um sistema especialista possui a base de regras, que constitui o conhecimento, a base de fatos e a máquina de inferência.

Como Jess nada mais é do que uma classe Java, a integração entre as duas linguagens é completa. Através de um programa escrito em Java, que utiliza os métodos de Jess, foi criada a base de regras do sistema especialista. Foi criado também um pacote escrito em Java, que contém uma classe para manipulação da *shell* Jess. Desse modo foi mais fácil trabalhar com Jess no momento em que a parte dinâmica do sistema estava sendo desenvolvida através da JSP.

Jess pode ser manipulado através de programas escritos em Java ou escritos em Jess, visto que, assim como Java, Jess também é uma linguagem de programação. Jess possui um *prompt* que aceita os comandos de sua linguagem, e que foi utilizado para testes, antes da implementação através da linguagem Java.

O sistema especialista implementado para este projeto, possui uma base de regras contendo 140 regras. Em tempo de execução a base de fatos é criada. A máquina de inferência esta implementada na *shell* Jess, através do algoritmo de Rete, precisando apenas executá-la para colocá-la em funcionamento.

Através do cadastro da amostra ambiental, o usuário atribui os valores para as variáveis do sistema especialista. São elas: temperatura, salinidade, direção do vento, intensidade do vento, transparência, estação do ano, ondas, correnteza, maré e coliformes. Quando o usuário pedir para o sistema calcular o diagnóstico da amostra, através do módulo amostra ambiental, a base de fatos será criada conforme dados previamente cadastrados pelo usuário. O sistema seleciona no banco de dados a amostra que o usuário escolheu para encontrar o diagnóstico e envia para o sistema especialista os valores de cada variável, que constituirão a base de fatos.

Com a base de fatos definida, o SE carrega as regras para constituir a base de regras. As regras foram armazenadas no arquivo “regras.dat”. Feito isto, a máquina de inferência é executada e o diagnóstico para a amostra é encontrado.

Importante ressaltar, que para encontrar o valor do diagnóstico, são utilizadas cinco variáveis intermediárias: condição1, condição2, condição3, condição4 e condição5. Cada variável possui seu valor (ótimo, bom, regular ou ruim). Para encontrar o diagnóstico, foi necessária a combinação entre essas cinco variáveis intermediárias.

O sistema *web* recupera no banco de dados a amostra selecionada pelo usuário através do módulo amostra ambiental e utiliza a classe que foi criada para manipulação do Jess através da linguagem Java, chamada “UtilizaJess”. Esta classe possui um método chamado “CalculaDiagnostico”, que recebe como parâmetros os valores para as variáveis do sistema especialista.

A base de regras para o sistema especialista foi desenvolvida utilizando um programa Java, que importa as classes de Jess necessárias para isto.

O primeiro passo para desenvolvimento do sistema especialista foi a criação da base de regras. Para definir uma regra, utiliza-se o comando de Jess “*defrule*”.

A classe UtilizaJess foi criada para facilitar a utilização de Jess, por um programa Java. Foi na construção da parte dinâmica do sistema *web* que encontrou-se esta necessidade. Os comandos de Jess necessários para utilizar o sistema especialista estão implementados nesta classe. Para poder utilizá-la através da JSP, optou-se por incluí-la num pacote de classes. Entre as vantagens da utilização de pacotes em Java, é que facilitam a reutilização de *software*, permitindo a importação de classes de outros pacotes, além de fornecerem uma convenção para nomes de classes. Isto ajuda, para que não exista conflito entre nomes de classes, devido à grande quantidade de programadores Java em todo mundo, cerca de centenas de milhares, segundo DEITEL & DEITEL (2003).

A interface com usuário, é composta da parte visual e parte dinâmica. A visual, foi desenvolvida através da ferramenta da Macromedia Dreamweaver 4.0. Tal ferramenta possui vários recursos que ajudam na criação de páginas para internet. As telas possuem um cabeçalho em comum, o qual possui os botões para navegação no *site*. Cada módulo, abre uma tabela contendo os *links* que levam o usuário ao módulo requisitado.

Os botões são importados para a página HTML gerada pelo Dreamweaver. Foram criados em outra ferramenta também da empresa Macromedia, o Flash 5.0. A página inicial possui fotos que foram tiradas do Parque de Cultivo da Enseada da Armação do Itapocoroí, Penha SC.

A Figura 1 apresenta a página inicial do sistema. O cabeçalho é incluído na página através de uma linha de comando em JSP. Logo abaixo, o *link* Página Inicial indica a localização do usuário no sistema, portanto o usuário está na página inicial. Esta página é bem simples, e seu objetivo é apresentar um ponto de partida para o sistema.

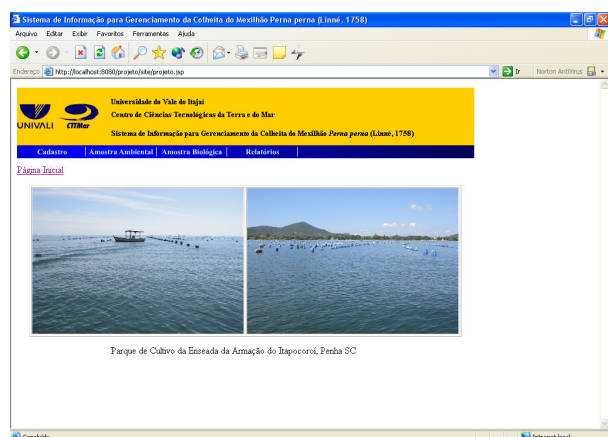


Figura 1: Página inicial do sistema.

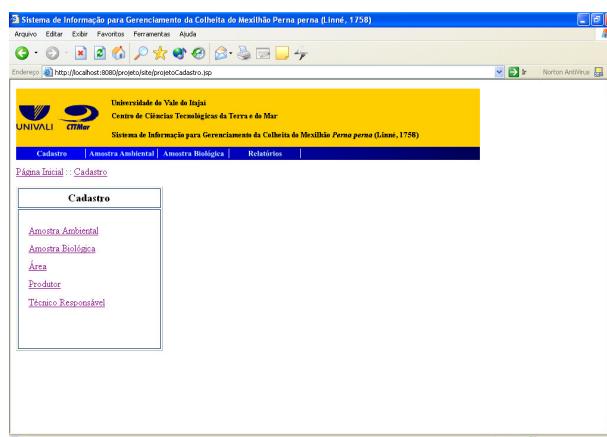


Figura 2: Página do módulo Cadastro.

A Figura 2 apresenta a página para o módulo de Cadastros do sistema. Além do *link* Página Inicial, esta página possui ao lado o *link* Cadastro. Sempre o último *link* indica a localização no site para o usuário. Cada *link* permitirá ao usuário acessar as páginas à que estes se referirem.

Esta página possui os *links* para os cadastros do sistema: Amostra Ambiental, Amostra Biológica, Área, Produtor e Técnico Responsável. A seguir, serão mostradas as telas que compõe o cadastro da amostra ambiental. Para todos cadastros, o *layout* e a funcionalidade são os mesmos, por isso as páginas dos demais cadastros do sistema não serão exibidas.

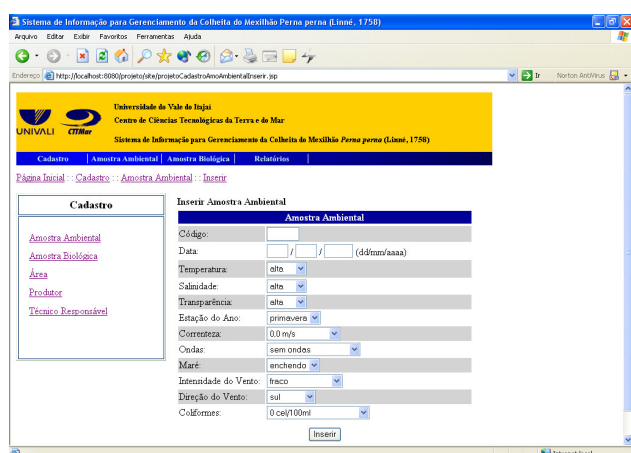


Figura 3: Página Inserir Amostra Ambiental.

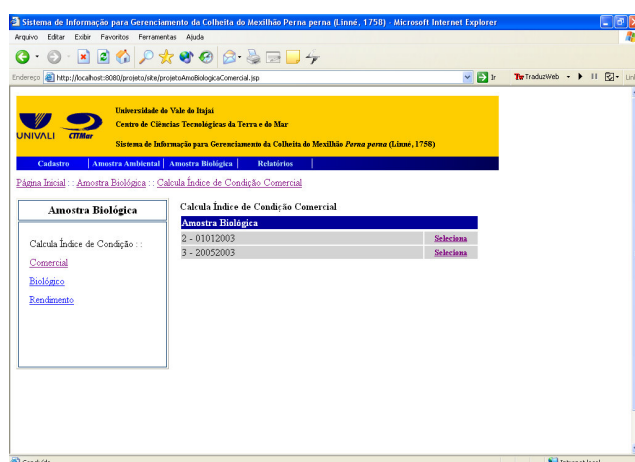


Figura 4. Cálculo do Índice de Colheita

3.2 Validação

Para validação do sistema, foi submetido ao professor responsável pelo Campus V da Univali, localizado no município de Penha-SC. Portanto, os módulos cadastro, amostra ambiental, amostra biológica e relatórios foram submetidos à sua utilização. Foram analisados além de aspectos funcionais, também aspectos ergonômicos da interface com o usuário, como a condução, carga de trabalho, controle explícito, adaptabilidade, gestão de erros, coerência, significado de códigos e compatibilidade.

Para isto, foi preparado um questionário, e para cada questão, existem quatro opções de resposta: ótimo, bom, regular, ruim. Sendo que, a Tabela 1 indica os procedimentos necessários para cada resposta do questionário e a Tabela 12 traz a validação para os módulos do sistema.

Tabela 1: Procedimentos necessários de acordo com a resposta da questão.

Resposta	Procedimento
Ótimo	Nenhum.
Bom	Precisa melhorar pouco.
Regular	Precisa melhorar.
Ruim	Precisa melhorar muito.

Tabela 1: Validação para os módulos do sistema.

Questões	Ótimo	Bom	Regular	Ruim
Nomes para os campos (<i>label</i>)		X		
Campos (Incluir ou Excluir)			X	
Condução (Mensagens, rótulos, alarmes)		X		
Carga de Trabalho (Eficiência do diálogo)	X			
Controle Explícito (Controle do sistema)		X		
Adaptabilidade (Várias maneiras para acesso)	X			
Gestão de Erros (Evitar ou reduzir erros)			X	
Coerência (<i>Layouts</i> idênticos em contextos iguais)	X			
Significado de Códigos (Relação semântica)	X			
Compatibilidade (Facilidade operação do sistema)	X			

Com relação à primeira questão, avaliou-se a necessidade de alterar o nome de alguns campos. Apenas o cadastro da área precisou de alterações, descritas na Tabela 3. Tais alterações contribuem para que o usuário tenha um melhor entendimento em relação à informação que deve ser impostada no campo.

Tabela 3: Alterações para módulo cadastro referentes questão ‘Nomes para os campos’.

Nome	Nome Alterado
Situação Geográfica	Localização Geográfica
Profundidade	Profundidade Média
Características do Cultivo	Tipo Cultivo
Tempo Cultivo	Tempo Total até Colheita

A questão referente aos campos avaliou se algum campo deveria ser incluído ou excluído. As alterações para o cadastro da área estão na Tabela 4. O campo excluído, não era necessário pois não afeta em nenhum aspecto a área de cultivo. Já os campos que foram incluídos, em relatórios mais elaborados, podem trazer informações importantes, podendo descobrir porque determinada área de cultivo esta produzindo mais ou menos do que uma outra área.

Tabela 4: Campos incluídos ou excluídos do cadastro das áreas.

Nome do campo	Incluído	Excluído
Exposição as ondas		X
Preço de venda p/kg	X	
Nº de empregados	X	
Tamanho médio das cordas de cultivo	X	
Número de cordas	X	

Para o cadastro das amostras biológicas, os campos peso da concha vazia e peso bruto, foram excluídos, por não serem utilizados pelas fórmulas que calculam os índices de condição comercial, biológico e de rendimento. O cadastro de pesquisadores, teve seu nome alterado para técnico responsável, por abranger um maior número de profissionais que venham a utilizar o sistema. Além disso, este cadastro, assim como o cadastro de produtores, tiveram incluídos o campo *e-mail* para facilitar a comunicação com estas pessoas.

Conclusões e Recomendações

O sistema *web* desenvolvido, realizou as melhoras que eram necessárias no SEDICM, tanto nos aspectos ergonômicos, quanto funcionais. A escolha da linguagem Java para programação foi ideal, pois possibilitou total integração com a *shell* Jess, e com as *tags* JSP. Este sistema não foi ainda publicado na internet, pois é fundamental criar um controle de acessos de usuários, assim como fazer um estudo aprofundado sobre as classes de Java que implementam a segurança para sistemas voltados à *web*.

Neste projeto, foram utilizadas tabelas do tipo MyISAM para o banco de dados. O problema dessas tabelas é que não permitem a integridade relacional entre elas. O banco de dados MySQL, dentre vários tipos, propõe o tipo InnoDB, tecnologia recente, que pode ser estudada e implementada neste projeto, pois este tipo de tabela resolve o problema da integridade relacional. Ainda em relação ao banco de dados, solução melhor, será utilização de um banco orientado a objetos, visto que a modelagem, em UML, e a linguagem de programação, em Java, são orientados a objetos.

Pode-se melhorar a utilização da JSP, otimizando a programação Java, através da utilização da JSTL (*JSP Standard Tag Library*). A JSTL é uma biblioteca de *tags* customizadas que estende o JSP,

não sendo necessário desenvolver código Java para operações comuns como formatação de texto ou acesso a bancos de dados.

Entre algumas modificações que poderão ser feitas, a criação de um cadastro para as instituições à que esta vinculado o técnico responsável, possibilitará excelente troca de informações. Não obstante, a criação de um cadastro de parques de cultivo, onde as áreas estariam vinculadas, permitiria uma possível abrangência a nível nacional do sistema, tendo em vista que a produção de mexilhões no estado de Santa Catarina é de 90 % da produção nacional, estando os 10 % restantes, divididos principalmente entre os estados do Espírito Santo, São Paulo e Rio de Janeiro.

Bibliografia

DEITEL, H. M.; DEITEL, P. J. **Java, como programar**. Tradução Carlos Arthur Lang Linboa. 4ª Edição, Porto Alegre, Bookman, 2003

FERNANDES, A.M.R. **Sistema Especialista Difuso Aplicado ao Processo de Análise Química Qualitativa de Amostras de Minerais**. Florianópolis, 1995. Dissertação (Mestrado no Curso de pós Graduação em Ciências da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina.

JESS, The Java Expert System Shell. Sandia National Laboratories. Disponível em <<http://herzberg.ca.sandia.gov/jess>> Acesso em 20 fev 2003.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de Informação com Internet**. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 1999.

LEITE, Jair C. **Notas de aula de Engenharia de Software**. Departamento de Informática e Matemática Aplicada UFRN. 2000.

LIEBOWITZ, J. **WorldWide Expert Systems Activities and Trends**. Cognizant Communications Offices, USA, 1994.

O'BRIEN, JAMES A. **Sistemas de Informação e as decisões gerenciais na era da Internet**. São Paulo: Saraiva, 2001.

SÁ, E. J. **SEDICM - Sistema Especialista Para A Determinação Do Índice De Condição De Colheita Do Mexilhão Perna perna (Linné, 1758)**. 1999. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade do Vale do Itajaí, Itajaí.

STAIR, R. M.; REYNOLDS, G. W. **Principles of Information Systems**. Fourth Edition, 1999, Course Technology. ISBN 0-7600-1079-X.

WHALEY, Charles. **Artificial Intelligence is part of daily life: Robotics, Auto-Focus Câmeras and Stock Market Use**. 01 mar 2002. Disponível em <<http://plesman.com/cc/home.html>> Acesso em 04 mar 2003.